

WHAT IS CLAIMED IS:

1. A networked system comprising:
 - a network;
 - a server coupled to the network, wherein the server includes:
 - an application code source that stores a client application; and
 - a server code manager coupled to the application code source; and
 - a client coupled to the network, wherein the client includes:
 - a CPU for natively executing code segments derived from the client application;
 - a code cache coupled to the CPU, for storing code segments; and
 - a client code manager coupled to the code cache, wherein the client code manager launches the client application by requesting that the server code manager transmit a code segment to the client, receiving the code segment from the server, storing the code segment in the code cache, and executing the code segment using the CPU until the code segment attempts to pass control to a required code segment not stored in the code cache, at which point control passes back to the client code manager to retrieve the required code segment from the server, with the CPU continuing execution with the required code segment.
2. The networked system of claim 1 wherein the server code manager comprises:
 - an application code transformation manager coupled to the application code source, for transforming the client application from a first format to a native binary format compatible with a native instruction set of the CPU of the client; and

5 a server code segment manager coupled to the application code transformation
6 manager, for parsing the client application in the native binary format into code
7 segments that are transmitted to the client via the network.

1 3. The networked system of claim 2 wherein the first format is a native binary format
2 other than the native binary format of the CPU of the client, and the application code
3 transformation manager comprises a transformation engine to transform the client application
4 from the first format to the native binary format of the CPU of the client.

5 4. The networked system of claim 2 wherein the first format is a source code text format
6 of a programming language, and the application code transformation manager comprises a
7 compiler that compiles and links the client application into a native binary format of the CPU of
8 the client.

1 5. The networked system of claim 2 wherein the first format is a virtual machine format,
2 and the application code transformation manager comprises a just-in-time compiler that compiles
3 and links the client application into a native binary format of the CPU of the client.

4 6. The networked system of claim 1 wherein the client code manager comprises:
5 a client code segment manager coupled to the network and the code cache, wherein
6 the client code manager requests needed segments from the server; and
7 a code cache linker and manager coupled to the code cache and client code cache
8 segment manager, wherein the code cache linker and manager links the code
segment received from the server into the code cache, emits the received code
segment into the code cache, and branches to the received code segment in the
code cache.

7. The networked system of claim 6 wherein the code cache linker and manager adjusts any branch targets in code segments stored in the code cache that need to branch to the received code segment and had previously been adjusted to branch out of the code cache to the client code segment manager to now branch to appropriate locations within the received code segment, adjusts any branch instructions in the received code segment having branch targets that branch to code segments currently in the code cache to branch to the appropriate code segments in the code cache, and adjust any branch instructions in the received code segment having branch targets that need to branch to code segments not in the code cache to branch out of the code cache to the client code segment manager to request the code segment containing the branch targets.

8. The networked system of claim 6 wherein the code cache linker and manager includes a code cache maintenance unit coupled to the code cache, for removing old and unneeded code segments from the code cache, replacing older segments with newly received segments when the code cache reaches a certain threshold, and optimizing code segments in the code cache.

9. A server comprising:
a network interface that couples the server to a network, which in turn is coupled to a client;
an application code source that stores a client application; and
a server code manager coupled to the application code source and the network interface, wherein the server code manager derives native binary code segments in a native execution format required by client CPUs from the application code source, and transmits the native binary code segments to clients upon requests received from clients.

10. The server of claim 9 wherein the server code manager comprises:

an application code transformation manager coupled to the application code source,
for transforming the client application from a first format to the native execution
format; and
a server code segment manager coupled to the application code transformation
manager, for parsing the client application in the native execution format into
code segments that are transmitted to clients via the network.

11. The server of claim 10 wherein the first format is a native execution format other than the native execution format required by a client CPU, and the application code transformation manager comprises a transformation engine to transform the client application from the first format to the native execution format required by the client CPU.

12. The server of claim 10 wherein the first format is a source code text format of a programming language, and the application code transformation manager comprises a compiler that compiles and links the client application into a native binary format required by a client CPU.

13. The server of claim 10 wherein the first format is a virtual machine format, and the application code transformation manager comprises a just-in-time compiler that compiles and links the client application into a native binary format required by a client CPU.

14. A client comprising:
a network interface that couples the client to a network, which in turn is coupled to a server;
a CPU for natively executing code segments derived from the client application stored on the server;
a code cache coupled to the CPU, for storing code segments; and

7 a client code manager coupled to the code cache, wherein the client code manager
8 launches the client application by requesting that the server transmit a code
9 segment to the client, receiving the code segment from the server, storing the
10 code segment in the code cache, and executing the code segment using the CPU
11 until the code segment attempts to pass control to a required code segment not
12 stored in the code cache, at which point control passes back to the client code
13 manager to retrieve the required code segment from the server, with the CPU
14 continuing execution with the required code segment.

15. The client of claim 14 wherein the client code manager comprises:
a client code segment manager coupled to the network and the code cache, wherein
the client code manager requests needed segments from the server; and
a code cache linker and manager coupled to the code cache and client code cache
segment manager, wherein the code cache linker and manager links the code
segment received from the server into the code cache, emits the received code
segment into the code cache, and branches to the received code segment in the
code cache.

16. The client of claim 15 wherein the code cache linker and manager adjusts any branch
targets in code segments stored in the code cache that need to branch to the received code
segment and had previously been adjusted to branch out of the code cache to the client code
segment manager to now branch to appropriate locations within the received code segment,
adjusts any branch instructions in the received code segment having branch targets that branch
to code segments currently in the code cache to branch to the appropriate code segments in the
code cache, and adjust any branch instructions in the received code segment having branch
targets that need to branch to code segments not in the code cache to branch out of the code

9 cache to the client code segment manager to request the code segment containing the branch
10 targets.

1 17. The client of claim 15 wherein the code cache linker and manager includes a code
2 cache maintenance unit coupled to the code cache, for removing old and unneeded code
3 segments from the code cache, replacing older segments with newly received segments when the
4 code cache reaches a certain threshold, and optimizing code segments in the code cache.

1 18. A method of executing an application in networked system comprising:
2 at a client:
3 issuing a code segment request to a server coupled to the client by a network;
4 at the server:
5 receiving the code segment request from the client;
6 deriving a code segment in a native execution format required by the client from
7 an application code source; and
8 transmitting the code segment to the client; and
9 at the client:
10 receiving the code segment;
11 adjusting branches in the code segment having targets not in a code cache of the
12 client to cause code segments containing the targets to be requested
13 from the server;
14 emitting the code segment into the code cache; and
15 executing the code segment natively from the code cache.

1 19. The method of claim 18 wherein deriving a code segment in a native execution format
2 required by the client from an application code source comprises transforming the application

3 source from a native execution format other than that required by the client into the native
4 execution format required by the client.

1 20. The method of claim 18 wherein deriving a code segment in a native execution format
2 required by the client from an application code source comprises compiling and linking the
3 application code source from a code text format of a programming language into the native
4 execution format required by the client.

1 21. The method of claim 18 wherein deriving a code segment in a native execution format
2 required by the client from an application code source comprises using a just-in-time compiler
3 to compile the application code source from a virtual machine format into the native execution
4 format required by the client.

1 22. The method of claim 18 wherein the code segment is a first code segment, and
2 executing the code segment natively from the code cache includes:

3 at the client:

4 executing a branch in the first code segment that seeks to branch to a second
5 code segment not in the code cache; and

6 issuing a code segment request for the second code segment to the server;

7 at the server:

8 receiving the code segment for the second code segment from the client;

9 deriving the second code segment in the native execution format required by the

10 client from the application code source; and

11 transmitting the second code segment to the client; and

12 at the client:

13 receiving the second code segment;

adjusting any branches in the first code segment stored in the code cache that need to branch to the second code segment and had previously been adjusted to cause the second code segment to be requested from the server to now branch to appropriate locations within the second code segment;

adjusting any branches in the second code segment having targets in the first code segment in the code cache to branch to the appropriate location within the first code segment;

adjusting any branches in the second code segment having branch targets in code segments not in the code cache to cause the code segments not in the code cache to be requested by the server;

emitting the second code segment into a code cache; and

continuing execution in the second code segment at the location to which the first code segment attempted to branch.

23. A computer program product, comprising:
- at least one computer usable medium having computer readable code embodied therein for causing an application to be executed in a networked system, the computer program product including:
- first computer readable program code devices configured to cause a client to issue a code segment request to a server coupled to the client by a network;
- second computer readable program code devices configured to cause the server to receive the code segment request from the client, derive a code segment in a native execution format required by the client from an application code source, and transmit the code segment to the client;
- and

third computer readable program code devices configured to cause the client to receive the code segment, adjust branches in the code segment having targets not in a code cache of the client to cause code segments containing the targets to be requested from the server, emit the code segment into the code cache, and execute the code segment natively from the code cache.

24. The computer program product of claim 23 wherein the code segment is a first code segment, and the third computer readable program code devices includes:

fourth computer readable program code devices configured to cause the client to execute a branch in the first code segment that seeks to branch to a second code segment not in the code cache, and issue a code segment request for the second code segment to the server;

fifth computer readable program code devices configured to cause the server to receive the code segment for the second code segment from the client, derive the second code segment in the native execution format required by the client from the application code source, and transmit the second code segment to the client; and

sixth computer readable program code devices configured to cause the client to receive the second code segment, adjusting any branches in the first code segment stored in the code cache that need to branch to the second code segment and had previously been adjusted to cause the second code segment to be requested from the server to now branch to appropriate locations within the second code segment, adjust any branches in the second code segment having targets in the first code segment in the code cache to branch to the appropriate location within the first code segment, adjust any branches in the second code segment having branch targets in code segments not in the code cache to cause

21 the code segments not in the code cache to be requested by the server, emit the
22 second code segment into a code cache, and continue execution in the second
23 code segment at the location to which the first code segment attempted to
24 branch.